PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-063355
(43)Date of publication of application : 08.03.1996

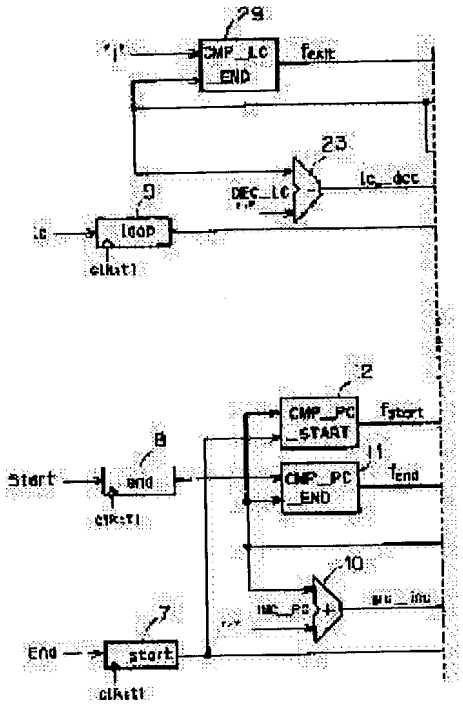
(51)Int.Cl. G06F 9/32

(21)Application number : 06-194037
(22)Date of filing : 18.08.1994

(71)Applicant : MITSUBISHI ELECTRIC CORP
(72)Inventor : SUTORAITENBERUGAA ROBAATO
KAWAI HIROYUKI
INOUE YOSHITSUGU

(54) PROGRAM CONTROLLER AND PROGRAM CONTROL METHOD

(57)Abstract:
PURPOSE: To provide a program controller which can effectively control the loop processing.
CONSTITUTION: The value of the start and end addresses of the loop processing are stored in a start register 7 and an end register 8 respectively and synchrnously with a clock t1. The data stored in both registers 7 and 8 are inputted to the comparators 12 and 11 respectively. The comparator 12 compares the output of a delay program counter with the data stored in the start register 7 and then sets a flag 'fstsrt' only when the coincidence is secured in the comparison. The flag 'fstart' is reset in other cases. Meanwhile the comparator 11 compares the output of the delay program counter with the data stored in the end register 8 and sets a flag 'fend' only when the coincidence is secured in the comparison. The flag 'fend' is reset in other cases.



LEGAL STATUS

[Date of request for examination] 24.08.2000

[Date of sending the examiner's decision of rejection] 04.11.2003

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12) **公開特許公報(A)**　(11)特許出願公開番号

特開平8－63355

(43)公開日　平成８年(1996)３月８日

(51)Int.Cl.[6]	識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０６Ｆ　9/32	３３０　Ａ Ｄ			

審査請求　未請求　請求項の数３　ＯＬ　（全 16 頁）

(21)出願番号　特願平6－194037

(22)出願日　平成６年(1994)８月18日

(71)出願人　000006013
三菱電機株式会社
東京都千代田区丸の内二丁目２番３号

(72)発明者　ストライテンベルガー　ロバート
兵庫県伊丹市瑞原４丁目１番地　三菱電機株式会社システムエル・エス・アイ開発研究所内

(72)発明者　河合　浩行
兵庫県伊丹市瑞原４丁目１番地　三菱電機株式会社システムエル・エス・アイ開発研究所内

(74)代理人　弁理士　吉田　茂明　（外２名）

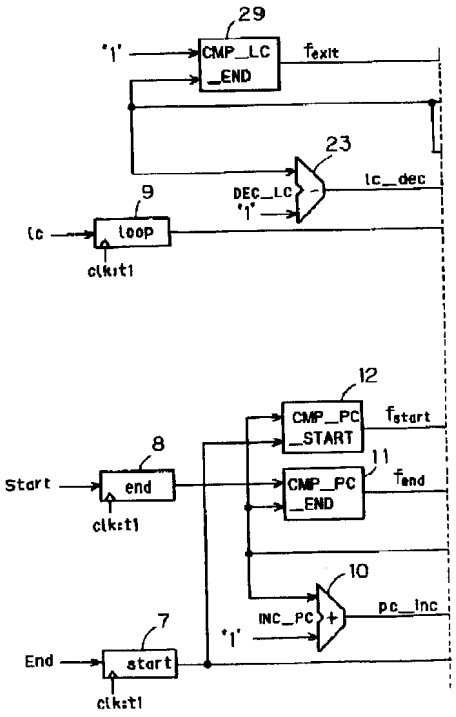
最終頁に続く

(54)【発明の名称】　プログラム制御装置及びプログラム制御方法

(57)【要約】

【目的】　効率的なループ処理が制御可能なプログラム制御装置を得る。

【構成】　ループ処理のＳｔａｒｔアドレス及びＥｎｄアドレスそれぞれの値は、クロックｔ１に同期して、レジスタ（ｓｔａｒｔ）７及びレジスタ（ｅｎｄ）８にそれぞれ記憶される。レジスタ７の格納データｓｔａｒｔとレジスタ８の格納データｅｎｄは、それぞれコンパレータ１２及びコンパレータ１１に入力される。コンパレータ１２は、遅延プログラムカウンタ１８の出力とデータｓｔａｒｔとを比較して、両者が等しいときのみフラグｆstartセットし、他の場合はリセットする。コンパレータ１１は、遅延プログラムカウンタ１８の出力とデータｅｎｄとを比較して両者が等しいときのみフラグｆendをセットし、他の場合はリセットする。



1

【特許請求の範囲】

【請求項１】　アドレスに対応した複数の命令からなるプログラムの実行制御を行うプログラム制御装置であって、前記プログラム制御装置は、ループ処理実行に際し、ループ処理の開始アドレス及び終了アドレス並びに反復数を規定したループ命令と、前記開始アドレスと前記終了アドレスとの間に記述された少なくとも１つの命令からなるループ実行命令群とを前記プログラム中に記述することを要求し、

現在の実行対象の命令のアドレスを規定するプログラムカウンタを所定時間間隔でインクリメントして出力するプログラムカウンタ出力手段と、

前記開始アドレスを格納する開始アドレス記憶手段と、

前記終了アドレスを格納する終了アドレス記憶手段と、

前記反復数を格納する反復数記憶手段と、

前記プログラムカウンタと前記開始アドレスとの比較結果に基づきループ処理の開始を認識し、前記プログラムカウンタと前記終了アドレスとの比較結果に基づきループ処理の終了を認識することにより、前記ループ実行命令群の命令が前記反復数回、繰り返して実行されるように前記プログラムカウンタを設定するループ制御手段とを備える、プログラム制御装置。

【請求項２】　アドレスに対応した複数の命令からなるプログラムの実行制御を行うプログラム制御装置であって、前記プログラム制御装置は、ネスティングされたループ処理の実行に際し、第１～第ｎ（ｎ≧２）の順にネスティングされた第１～第ｎループ処理それぞれの開始アドレス及び終了アドレス並びに反復数を規定した第１～第ｎのループ命令と、前記第１～第ｎのループ命令それぞれで規定された前記開始アドレスと前記終了アドレスとの間に記述された少なくとも１つの命令からなる第１～第ｎのループ実行命令群とを前記プログラム中に記述することを要求し、

現在の実行対象の命令のアドレスを規定するプログラムカウンタを所定時間間隔でインクリメントして出力するプログラムカウンタ出力手段と、

前記第１～第ｎのループ処理の前記開始アドレスを第１～第ｎの開始アドレスとして格納する第１～第ｎの開始アドレス記憶手段と、

前記第１～第ｎのループ処理の前記終了アドレスを第１～第ｎの終了アドレスとして格納する第１～第ｎの終了アドレス記憶手段と、

前記第１～第ｎのループ処理の前記反復数を第１～第ｎの反復数として格納する第１～第ｎの反復数記憶手段と、

選択制御信号に基づき、前記第１～第ｎの開始アドレスのうちの一のアドレスを選択開始アドレスとして出力する開始アドレス選択手段と、

前記選択制御信号に基づき、前記第１～第ｎの終了アドレスのうちの一のアドレスを選択終了アドレスとして出力する終了アドレス選択手段と、

2

前記選択制御信号に基づき、前記第１～第ｎの反復数のうちの一の反復数を選択反復数として出力する反復数選択手段と、

前記プログラムカウンタと前記選択開始アドレスとの比較結果及び前記プログラムカウンタと前記選択終了アドレスとの比較結果に基づき、現在制御対象となるループ処理を指示する選択制御信号を出力する選択制御信号出力手段と、

前記プログラムカウンタと前記選択開始アドレスとの比較結果に基づき、選択開始アドレスに対応するループ処理の開始を認識し、前記プログラムカウンタと前記選択終了アドレスとの比較結果に基づき前記選択終了アドレスに対応するループ処理の終了を認識して、制御対象のループ処理が前記選択反復数回、繰り返して実行されるように前記プログラムカウンタを設定するループ制御手段とを備える、プログラム制御装置。

【請求項３】　アドレスに対応した複数の命令からなるプログラムの実行制御を行うプログラム制御方法であって、前記プログラム制御方法は、ループ処理実行に際し、ループ処理の開始アドレス及び終了アドレス並びに反復数を規定したループ命令と、前記開始アドレスと終了アドレスとの間に記述された少なくとも１つの命令からなるループ実行命令群とを前記プログラム中に記述することを要求し、

(a) 現在の実行対象の命令のアドレスを規定するプログラムカウンタを前記プログラムの先頭アドレスから所定時間間隔でインクリメントするステップと、

(b) 前記プログラムカウンタの指示するアドレスの命令がループ命令である場合にループ処理の開始アドレス、終了アドレス及び反復数を記憶するステップと、

(c) 前記プログラムカウンタと前記ステップ(b)で記憶した前記開始アドレスとの比較結果に基づきループ処理の開始を認識し、前記プログラムカウンタと前記ステップ(b)で記憶した前記終了アドレスとの比較結果に基づき前記ループ処理の終了を認識することにより、前記ループ実行命令群の命令が前記反復数回、繰り返して実行されるように前記プログラムカウンタを設定するステップとを備える、プログラム制御方法。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】この発明は、プログラムを実行制御するプログラム制御装置及びプログラム制御方法に関する。

【０００２】

【従来の技術】リアルタイムアプリケーションでは、命令がスムーズに流れることが重要である。例えば、任意の種類のプロセッサでフィルターアルゴリズムを実行する場合、個々のループ命令の繰り返しによる割り込みなしに垂直方向および水平方向の走査が行われると、適用

されたマトリクスの重複走査の結果は、速度の点からみると最高の結果が得られる。

【０００３】したがって、プログラムの実行を制御するプログラム制御装置は、通常、繰り返し処理を簡単に行うべく、ループ命令の実行を可能にしている。

【０００４】シングルループの場合、標準ループ命令の構文は図１７のようになる。なお、図１７において、“１ｃ”はループカウンタ、“Ｅｎｄ”は繰り返し実行されるループブロックの最終命令のアドレスを示す。

【０００５】当該命令にはプログラマにとって厳しい制限が１つある。すなわち、ループ命令は、ループブロックの第１行目（図１７の４０行）の直前の行（図１７の３９行）に記述しなければならない。

【０００６】ネスティングされた２つのループの標準的なプログラム記述例を図１８に示す。

【０００７】図１８から分かるように、３６行～４５行に記述されたＢｌｏｃｋ１のループブロックが、４０～４２行に記述されたＢｌｏｃｋ２のブロックを取り囲んでいる。Ｂｌｏｃｋ１のループブロックが実行される毎に、内側のＢｌｏｃｋ２のループ命令（３９行目）が繰返し実行される。

【０００８】３５行、３９行目にそれぞれ記述されたループ命令は、情報（ループカウンタ、開始および終了アドレス）の伝達を行うための命令であるため、１度実行すれば十分である。

【０００９】しかしながら、Ｂｌｏｃｋ２のループ命令は、Ｂｌｏｃｋ１のループが繰り返される度に実行されるため、（ｌｃ１－１）回余分に実行されることになり、非効率的な実行となる。この非効率な実行により、フィルターアプリケーション等の実行の際には大きな悪影響を受ける。以下、その詳細な説明を行う。

【００１０】フィルターアプリケーションのプログラムを実行する際、画素データのマトリクスが３対３要素の重畳マスク（例えば、“ラプラシアン端線検出”）を付与される場合、各水平方向走査線の開始と終了で特別の処理を行わなければならない。境界画素の外挿のため、図１９に示すように、“仮想の”ゼロを境界に沿って挿入する。

【００１１】フレームの内部画素は、インデックス配列を用いて容易に処理され、データスタックのアドレス指定が可能となる。マスクがシフトされると、インデックスはインクリメントされる。よって、マスクの水平方向走査のために、繰返しブロック、すなわちループ処理を行うことができる。しかし、マスクの境界に関しては、ゼロ挿入のためにブロックは不規則である。

【００１２】これによって、図１９で示した画素をマスク処理するには、処理される各フレームラインの開始と終了で、図２０に示すように、別個のブロック（ＢｌｏｃｋＡ，ＢｌｏｃｋＢ，ＢｌｏｃｋＣ）を用いてプログラムを作成する必要がある。なお、図２０は１水平方向走査処理のプログラムであり、ａ［０］～ａ［５］に画素１～画素６の画素データが格納されるインデックス配列である。ａ［ｉ，１］は配列ａ［ｉ］の値を取り出した後に変数ｉを１インクリメントすることを意味し、ａ［ｉ，－１］は配列ａ［ｉ］の値を取り出した後に変数ｉを１デクリメントすることを意味している。ｗｒ［０］～ｗｒ［２］はマスクの第１行目の値を格納した配列であり、ｌｏａｄはデータロード命令、ｍｕｌａｃｃは演算命令である。また、図２０では示していないが、変種ｉの初期値は“１”である。

【００１３】標準ループ命令は、ＢｌｏｃｋＡとループブロックＢｌｏｃｋＢとの間に位置しなければならない（図２０の３９行目）。この３９行目のループ命令が不規則性を引き起こす。なぜなら、ＢｌｏｃｋＢを処理するまでに行われる工程は、境界画素を処理するＢｌｏｃｋＡの３工程に加え、ループ命令を含めると４工程が必要となり１工程増加するため、１水平方向走査のマスク処理を３工程で行えない箇所が出現して完全なフレームのリアルタイム処理が禁止されるからである。

【００１４】さらに、実際のフィルタリング処理は、上記水平方向走査を内部ループとし、外部ループ（垂直方向走査）が実行されるため、外部ループが繰り返される度に水平方向走査のループ命令を繰返さなければならないために、全体の処理時間が長くなるという問題点があった。

【００１５】

【発明が解決しようとする課題】従来のプログラム制御装置により、ループ命令は以上のように制御されるため、非効率であり、フィルタ処理等のアプリケーションの処理時間を長くしてしまうという問題点があった。

【００１６】この発明は上記問題点を解決するためになされたもので、効率的なループ処理が制御可能なプログラム制御装置及びプログラム制御方法を得ることを目的とする。

【００１７】

【課題を解決するための手段】この発明に係る請求項１記載のプログラム制御装置は、アドレスに対応した複数の命令からなるプログラムの実行制御を行う装置であって、前記プログラム制御装置は、ループ処理実行に際し、ループ処理の開始アドレス及び終了アドレス並びに反復数を規定したループ命令と、前記開始アドレスと前記終了アドレスとの間に記述された少なくとも１つの命令からなるループ実行命令群とを前記プログラム中に記述することを要求し、現在の実行対象の命令のアドレスを規定するプログラムカウンタを所定時間間隔でインクリメントして出力するプログラムカウンタ出力手段と、前記開始アドレスを格納する開始アドレス記憶手段と、前記終了アドレスを格納する終了アドレス記憶手段と、前記反復数を格納する反復数記憶手段と、前記プログラムカウンタと前記開始アドレスとの比較結果に基づきル

ープ処理の開始を認識し、前記プログラムカウンタと前記終了アドレスとの比較結果に基づきループ処理の終了を認識することにより、前記ループ実行命令群の命令が前記反復数回、繰り返して実行されるように前記プログラムカウンタを設定するループ制御手段とを備えて構成される。

【００１８】この発明に係る請求項２記載のプログラム制御装置は、アドレスに対応した複数の命令からなるプログラムの実行制御を行う装置であって、前記プログラム制御装置は、ネスティングされたループ処理の実行に際し、第１～第ｎ（ｎ≧２）の順にネスティングされた第１～第ｎループ処理それぞれの開始アドレス及び終了アドレス並びに反復数を規定した第１～第ｎのループ命令と、前記第１～第ｎのループ命令それぞれで規定された前記開始アドレスと前記終了アドレスとの間に記述された少なくとも１つの命令からなる第１～第ｎのループ実行命令群とを前記プログラム中に記述することを要求し、現在の実行対象の命令のアドレスを規定するプログラムカウンタを所定時間間隔でインクリメントして出力するプログラムカウンタ出力手段と、前記第１～第ｎのループ処理の前記開始アドレスを第１～第ｎの開始アドレスとして格納する第１～第ｎの開始アドレス記憶手段と、前記第１～第ｎのループ処理の前記終了アドレスを第１～第ｎの終了アドレスとして格納する第１～第ｎの終了アドレス記憶手段と、前記第１～第ｎのループ処理の前記反復数を第１～第ｎの反復数として格納する第１～第ｎの反復数記憶手段と、選択制御信号に基づき、前記第１～第ｎの開始アドレスのうちの一のアドレスを選択開始アドレスとして出力する開始アドレス選択手段と、前記選択制御信号に基づき、前記第１～第ｎの終了アドレスのうちの一のアドレスを選択終了アドレスとして出力する終了アドレス選択手段と、前記選択制御信号に基づき、前記第１～第ｎの反復数のうちの一の反復数を選択反復数として出力する反復数選択手段と、前記プログラムカウンタと前記選択開始アドレスとの比較結果及び前記プログラムカウンタと前記選択終了アドレスとの比較結果に基づき、現在制御対象となるループ処理を指示する選択制御信号を出力する選択制御信号出力手段と、前記プログラムカウンタと前記選択開始アドレスとの比較結果に基づき、選択開始アドレスに対応するループ処理の開始を認識し、前記プログラムカウンタと前記選択終了アドレスとの比較結果に基づき前記選択終了アドレスに対応するループ処理の終了を認識して、制御対象のループ処理が前記選択反復数回、繰り返して実行されるように前記プログラムカウンタを設定するループ制御手段とを備えて構成される。

【００１９】この発明に係る請求項３記載のプログラム制御方法は、アドレスに対応した複数の命令からなるプログラムの実行制御を行う方法であって、前記プログラム制御方法は、ループ処理実行に際し、ループ処理の開始アドレス及び終了アドレス並びに反復数を規定したループ命令と、前記開始アドレスと終了アドレスとの間に記述された少なくとも１つの命令からなるループ実行命令群とを前記プログラム中に記述することを要求し、(a) 現在の実行対象の命令のアドレスを規定するプログラムカウンタを前記プログラムの先頭アドレスから所定時間間隔でインクリメントするステップと、(b) 前記プログラムカウンタの指示するアドレスの命令がループ命令である場合にループ処理の開始アドレス、終了アドレス及び反復数を記憶するステップと、(c) 前記プログラムカウンタと前記ステップ(b)で記憶した前記開始アドレスとの比較結果に基づきループ処理の開始を認識し、前記プログラムカウンタと前記ステップ(b)で記憶した前記終了アドレスとの比較結果に基づき前記ループ処理の終了を認識することにより、前記ループ実行命令群の命令が前記反復数回、繰り返して実行されるように前記プログラムカウンタを設定するステップとを備えて構成される。

【００２０】

【作用】この発明における請求項１記載のプログラム制御装置のループ制御手段は、プログラムカウンタと開始アドレスとの比較結果に基づきループ処理の開始を認識することができるため、プログラム中においてループ命令はループ実行命令群の前であれば任意の位置に記述することができる。

【００２１】この発明における請求項２記載のプログラム制御装置は、プログラムカウンタと選択開始アドレスとの比較結果及びプログラムカウンタと選択終了アドレスとの比較結果に基づき、現在制御対象となるループ処理を指示する選択制御信号を出力する選択制御信号出力手段と、プログラムカウンタと選択開始アドレスとの比較結果に基づき、選択開始アドレスに対応するループ処理の開始を認識し、プログラムカウンタと選択終了アドレスとの比較結果に基づき選択終了アドレスに対応するループ処理の終了を認識して、制御対象のループ処理が前記選択反復数回、繰り返して実行されるようにプログラムカウンタを設定するループ制御手段とを備えている。

【００２２】したがって、プログラム中において、第１～第ｎのループ命令はそれぞれ第１～第ｎのループ実行命令群の前であれば任意の位置に記述することができる。

【００２３】この発明における請求項３記載のプログラム制御方法のステップ(c)は、プログラムカウンタと開始アドレスとの比較結果に基づきループ処理の開始を認識し、プログラムカウンタと終了アドレスとの比較結果に基づきループ処理の終了を認識することにより、ループ実行命令群の命令が反復数回、繰り返して実行されるようにプログラムカウンタを設定している。

【００２４】したがって、プログラム中においてループ

命令はループ実行命令群の前であれば任意の位置に記述することができる。

【００２５】

【実施例】

＜第１の実施例＞図１はこの発明の第１の実施例であるプログラム制御装置で実行制御可能なプログラムリストを示す説明図である。図１は、従来例で示した図１７に対応するシングルループ命令の例を示している。

【００２６】第１の実施例のプログラム制御装置は、開始アドレスと終了アドレスとを指定してループブロックを指示するループ命令を記述可能にする。したがって、このループ命令は、指示するループブロックより前であれば記述箇所が制限されない。図１の例では、アドレス４０～４２番に記述されたループブロックＢ１より、２０番前のアドレス２１番にループ命令「ｌｏｏｐ　１ｃ，Ｓｔａｒｔ，Ｅｎｄ」を記述している。

【００２７】図１に示すように、ループ命令の開始アドレスＳｔａｒｔと終了アドレスＥｎｄとが独立的に割当てられている。

【００２８】図１と図１７とを比較した場合、全体の処理時間はシングルループ命令に影響されないが、リアルタイム処理に必要な規則性は維持される。

【００２９】これによって、いかなるブロック区分されたアルゴリズムでもスムーズに処理可能になる。

【００３０】図１５は、第１の実施例のプログラム制御装置を含んだプログラム実行システムの全体構成を示す説明図である。同図に示すように、記憶装置１００内に複数の命令がアドレス（図１の行番号に相当）に対応して記述されたプログラム１０１が格納されており、プログラム制御装置１０２は、プログラム１０１の先頭アドレスからプログラムカウンタＰＣを所定時間間隔でインクリメントて命令実行装置１０３に出力する。この際、ループ処理を認識した場合はループ処理が所定回数繰り返されるようにプログラムカウンタＰＣを再設定する。命令実行装置１０３はプログラムカウンタＰＣで指示されたアドレスの命令を実行する。

【００３１】図２及び図３は、第１の実施例のプログラム制御装置の内部構成を示す回路図である。第１の実施例のプログラム制御装置は図１で示したシングルループ命令の実行を可能にする。図４～図６は動作説明用の信号波形図である。図４～図６において、波形を小さくかつ簡単にしておくため、２回の反復のみ選択した。また、２つの重複しないクロックｔ１とｔ２とを最上段に示している。なお、図４～図６の波形図は、図７のプログラムの実行状況を示している。図７のプログラムは従来例で示した図２０に対応している。

【００３２】レジスタ（ｔ２＿ＰＣ）１７及びレジスタ（ｔ１＿ＰＣ）１８はそれぞれクロックｔ１及びｔ２に同期してプログラム制御装置のプログラムカウンタＰＣをラッチする。なお、レジスタ１７から出力されるプログラムカウンタＰＣが命令実行装置１０３に出力される。

【００３３】図７で示したプログラムリストは行番号“２０”から始まるが、ここで反復数を指示するループ変数ｌｃには“２”の値がロードされる。

【００３４】本プログラムはパイプライン方式で実行されていない場合、ロード命令はクロックｔ２の次のサイクルで効果を現す。

【００３５】従って、ループ変数ｌｃを格納するレジスタ（ｌｏｏｐ）９はクロックｔ１に同期して“２”を格納して、ループブロックＢｌｏｃｋＢに関する反復数を記憶する。レジスタ９の出力信号はセレクタ（ＳＥＬ＿ＬＣ）２４に与えられており、このセレクタ２４は、可変ループカウンタ（ＬＣ）２６をクロックｔ２に同期して駆動し、可変ループカウンタ２６の出力がレジスタ（ｔ１＿ｌｃ）２８によりクロックｔ１に同期して格納される。

【００３６】レジスタ２８の出力は、デクリメンタ（ＤＥＣ＿ＬＣ）２３、コンパレータ（ＣＭＰ＿ＬＣ＿ＳＴＡＲＴ）３５およびコンパレータ（ＣＭＰ＿ＬＣ＿ＥＮＤ）２９に与えられる。

【００３７】コンパレータ２９は、レジスタ２８に格納されるループカウント値が“１”に等しい時にのみフラグｆexitをセット（Ｈ（“１”）に設定）し他の場合はリセット（Ｌ（“０”）に設定）する。コンパレータ３５は、レジスタ２８に格納されるループカウント値が“０”に等しい時はフラグｆlcをセットし、他の場合はリセットする。

【００３８】デクリメンタ２３によりデクリメントされたループカウント値ｌｃ＿ｄｅｃは、セレクタ（ＳＥＬ＿ＬＣ）２４の第２入力となる。

【００３９】制御信号ｆsel_lcは論理回路２０によって生成されるが、これについては後述する。

【００４０】図７で示したプログラムの行番号２１において、ループ命令が付与される。このループ命令には、ループブロックの開始と終了に関する情報が含まれている。

【００４１】デバッグ時の取扱い易さとプログラムの読み易さのため、絶対アドレスの代わりにラベルが付けられている。

【００４２】図７の場合、ラベル“Ｓｔａｒｔ”は行番号４０に対応し、ラベル“Ｅｎｄ”は行番号４２に対応する。

【００４３】ＳｔａｒｔアドレスおよびＥｎｄアドレスそれぞれの値は、クロックｔ１に同期して、レジスタ（ｓｔａｒｔ）７及びレジスタ（ｅｎｄ）８にそれぞれ記憶される。

【００４４】遅延プログラムカウンタ１８の出力は、インクリメンタ（ＩＮＣ＿ＰＣ）１０、コンパレータ（ＣＭＰ＿ＰＣ＿ＳＴＡＲＴ）１２及びコンパレータ（ＣＭ

9

P_PC_END）１１に付与される。

【００４５】インクリメンタ１０の出力信号ｐｃ＿ｉｎｃとレジスタ７の格納データｓｔａｒｔはセレクタ（ＳＥＬ＿ＰＣ）１５の入力となり、このセレクタ１５の出力はプログラムカウンタ１７を駆動する。

【００４６】レジスタ７の格納データｓｔａｒｔとレジスタ８の格納データｅｎｄは、それぞれコンパレータ１２及びコンパレータ１１に入力される。コンパレータ１２は、遅延プログラムカウンタ１８の出力とデータｓｔａｒｔとを比較して、両者が等しいときのみフラグｆｓｔａｒｔセットし、他の場合はリセットする。コンパレータ＊

10

＊１１は、遅延プログラムカウンタ１８の出力とデータｅｎｄとを比較して両者が等しいときのみフラグｆｅｎｄをセットし、他の場合はリセットする。

【００４７】セレクタ（ＳＥＬ＿ＰＣ）１５への選択制御フラグｆｓｅｌ_ｐｃは、フラグｆｅｎｄ及びｆｅｘｉｔを入力とする論理回路（Ｌｏｇｉｃ＿２）１９によって生成される。この論理回路１９の制御体系及びセレクタ１５の選択状況（ｓｅｌｅｃｔ１）を現す真理値表を表１に示す。

【００４８】

【表１】

fend	fexit	fsel_pc	Select1
0	?	0	pc_inc
1	0	1	start
1	1	0	pc_inc

？：don't care

【００４９】表１に示すように、フラグｆｅｎｄが“０”ときは、フラグｆｅｘｉｔの値に関係なく、選択制御フラグｆｓｅｌ_ｐｃを“０”にして、セレクタ１５によりインクリメンタ１０の出力信号ｐｃ＿ｉｎｃを選択させる。また、フラグｆｅｎｄが“１”ときにフラグｆｅｘｉｔが“０”の場合に選択制御フラグｆｓｅｌ_ｐｃを“１”にして、セレクタ１５によりレジスタ７の格納データｓｔａｒｔを選択させ、フラグｆｅｘｉｔが“１”の場合に選択制御フラグｆｓｅｌ_ｐｃを“０”にして、セレクタ１５によりインクリ※

※メンタ１０の出力信号ｐｃ＿ｉｎｃを選択させる。

【００５０】一方、セレクタ２４への選択制御信号ｆｓｅｌ_ｌｃは、フラグｆｓｔａｒｔ、ｆｅｎｄ及びｆｌｃを入力する論理回路（Ｌｏｇｉｃ＿１）２０によって生成される。この論理回路２０の制御体系及びセレクタ２４の選択状況（ｓｅｌｅｃｔ２）を示す真理値表を表２に示す。

【００５１】

【表２】

fstart	flc	fend	fsel_lc	Select2
default			00	lc
?	?	1	01	lc_dec
1	1	0	10	loop

？：don't care

【００５２】表２に示すように、デフォルト値として選択制御信号ｆｓｅｌ_ｌｃを“００”にして、セレクタ２４によりレジスタ２８の出力ｌｃを選択させる。また、フラグｆｅｎｄが“１”のときは、フラグｆｓｔａｒｔ、ｆｌｃに関係なく、選択制御信号ｆｓｅｌ_ｌｃを“０１”にして、セレクタ２４よりデクリメンタ２３の出力ｌｃ＿ｄｅｃを選択させる。さらに、フラグｆｅｎｄが“０”で、フラグｆｓｔａｒｔ、ｆｌｃが“１”のときは、選択制御信号ｆｓｅｌ_ｌｃを“１０”にして、セレクタ２４よりレジスタ９の格納データｌｏｏｐを選択させる。

【００５３】プログラムが行番号３７に達するまで、プログラムカウンタは、インクリメンタ１０の出力により常にインクリメントされる。また、各プログラムの開始時にループカウンタがリセットされるので、フラグ“ｆｌｃ”は既にセットされている。

【００５４】期間Ｔ１において、プログラムカウンタＰＣ＝“４０”の時、フラグｆｓｔａｒｔはクロックｔ１に同期してセットされる。フラグｆｓｔａｒｔ及びｆｌｃがセットされ、フラグｆｅｎｄがリセットされているので、表２で示した制御体系に基づき、フラグｆｓｅｌ_ｌｃは“１０”にセットされ、セレクタ２４はレジスタ９の格納データｌｏｏｐを選択する。

【００５５】したがって、可変ループカウンタ２６にはレジスタ９の格納データｌｏｏｐである“２”がロードされる。そして、１サイクル後にすべてのフラグがリセットされる。

【００５６】期間Ｔ２において、プログラムカウンタＰＣ＝“４２”の時、フラグｆｅｎｄはクロックｔ１に同期

してセットされる。

【００５７】フラグｆexitはリセットされるいるため、論理回路１９は表１の制御体系に従い、フラグｆsel_pcをセットし、セレクタ１５によりデータｓｔａｒｔが選択され、プログラムカウンタ１７及び１８はレジスタ７から格納データｓｔａｒｔの“４０”をロードする。

【００５８】同時に、可変ループカウンタ２６はデクリメンタ２３によりデクリメントされた値“１”を、セレクタ２４を介してロードし、その結果、コンパレータ２９によりフラグｆexitはセットされる。

【００５９】そして、期間Ｔ３において、プログラムカウンタＰＣ＝“４０”の時、フラグｆstartは再びセットされる。ｆstartの最初のセットと対照すると、フラグｆlcはリセットされているため、その結果、フラグ“ｆsel_lc”はリセットされる。

【００６０】その後、期間Ｔ４において、再びプログラムカウンタＰＣ＝“４２”の時、フラグｆendは再びセットされる。このとき、フラグ“ｆexit”は既にセットされているので、論理回路１９の選択制御フラグｆsel_pcはセットされない。

【００６１】したがって、セレクタ１５によりインクリメンタ１０の出力が選択されることにより、プログラムカウンタＰＣは連続的にインクリメントされる。一方、ループカウンタ２６にはそのデクリメンタ２３によりデクリメントされた値“０”をロードされる。

【００６２】その結果、コンパレータ３５によりフラグｆlcは強制的にセットされ、コンパレータ２９によりフラグｆexitはリセットされる。

【００６３】このように第１の実施例のプログラム制御装置は、プログラムカウンタとループの開始アドレスとの比較結果に基づきループ処理の開始を認識することができるため、プログラム中においてループ命令はループ実行命令群の前であれば任意の位置に記述することができる。

【００６４】例えば、図７で示したように、ループ実行命令群であるループブロックＢ［ｉ］のループ命令「ｌｏｏｐ　ｌｃ，Ｓｔａｒｔ，Ｅｎｄ」をブロックＢｌｏｃｋＡよりも前の行番号２１に記述することができる。

【００６５】したがって、図２０で示したプログラムのように、ブロックＢｌｏｃｋＡとループブロックＢｌｏｃｋＢとの間にループ命令を記述することによる不具合を解消することができる。なぜなら、ループブロックＢｌｏｃｋＢを処理するまでに行われる工程は、境界画素を処理するＢｌｏｃｋＡの３工程のみになり、ループブロックＢｌｏｃｋＢの１ループサイクル及びブロックＢｒｏｃｋＣも３工程で行えるため、１水平方向走査のマスク処理をすべて３工程で行うことができ、完全なフレームのリアルタイム処理が実現できるからである。

【００６６】すなわち、ループ命令をループ処理の実行の妨げにならない箇所に記述することにより、効率的なループ処理を行うことができる。

【００６７】図１６は、第１の実施例のプログラム制御装置と等価なプログラム制御方法を示すフローチャートである。

【００６８】同図を参照して、ステップＳ１で、ループ処理の開始アドレス及び終了アドレス並びに反復数を格納するレジスタを規定したループ命令と、開始アドレスと終了アドレスとの間に格納された少なくとも１つの命令からなるループ実行命令群とからなるループ処理を含むプログラムを読み込む。

【００６９】そして、ステップＳ２において、ステップＳ１で読み込んだプログラムの先頭アドレスにプログラムカウンタを設定する。

【００７０】次に、ステップＳ３において、プログラムカウンタの指示するアドレスの命令がループ命令であれば、ループ処理の開始アドレス、終了アドレス及び反復数を認識する。

【００７１】そして、ステップＳ４において、プログラムカウンタとステップＳ３で記憶した開始アドレスとの比較結果に基づきループ処理の開始を認識し、プログラムカウンタとステップＳ３で記憶した終了アドレスとの比較結果に基づきループ処理の終了を認識し、必要があればループ実行命令群の命令が反復数回、繰り返して実行されるようにプログラムカウンタにループ開始アドレスを再設定する。

【００７２】次に、ステップＳ５で現在のプログラムカウンタを命令実行装置に出力する。

【００７３】そして、ステップＳ６で、プログラムカウンタがプログラムの終了アドレスを指示しているか否かをＹＥＳ／ＮＯで判定し、ＹＥＳであれば終了しＮＯであればステップＳ７に移行する。

【００７４】ステップＳ７において、プログラムカウンタをインクリメントし、ステップＳ３に戻る。以降、ステップＳ６でＹＥＳと判定されるまで、ステップＳ３～Ｓ７が繰り返される。

【００７５】このように、このプログラム制御方法は、ステップＳ４で、プログラムカウンタと開始アドレスとの比較結果に基づきループ処理の開始を認識し、プログラムカウンタと終了アドレスとの比較結果に基づきループ処理の終了を認識することにより、ループ実行命令群の命令が反復数回、繰り返して実行されるようにプログラムカウンタを再設定することができる。

【００７６】したがって、プログラム中においてループ命令はループ実行命令群の前であれば任意の位置に記述することができる。

【００７７】その結果、ループ命令をループ処理の実行の妨げにならない箇所に記述することにより、効率的なループ処理を行うことができる効果を奏する。

【００７８】なお、図１６で示したプログラム制御方法は、図２及び図３で示した構成のプログラム制御装置以

外の構成でも実行可能であり、ソフトウェア処理で実行することも可能であることは勿論である。

【００７９】＜第２の実施例＞図８は第２の実施例のプログラム制御装置でプログラム制御可能プログラムを示す説明図である。図８は従来例で示した図１８に対応する２つのループブロックＢ１１，Ｂ１２のネスティングを示している。

【００８０】同図に示すように、行番号２１，２３で記述したループ命令で、ループ境界の開始アドレス（Ｓｔａｒｔ＿1，Ｓｔａｒｔ＿2）及び終了アドレス（Ｅｎｄ＿1，Ｅｎｄ＿2）を独立的に指定している。

【００８１】このようにネスト構造のループ命令を実行する際においても、ループ命令自体をループブロック前の任意の場所に記述することができる。

【００８２】第２の実施例のプログラム制御装置は、繰返しループ命令による割込みなしに、ネストされたループを実行することができ、それによって、プログラムの性能と効率を高めるとともに、リアルタイム処理を可能にしている。

【００８３】柔軟性を維持するために、ループカウンタの順序は制約されない。すなわち、図８で記述されたプログラムにおいて、ループブロックＢ１２も外部ループになりうる。

【００８４】図９及び図１０は、第２の実施例のプログラム制御装置の内部構成を示す回路図である。第２の実施例のプログラム制御装置は図８で示したネスティングループの実行を可能にする。図１１～図１３は動作説明用の波形図である。なお、図１１～図１３の波形図は、図１４のプログラムの実行状況を示している。波形を読み易くするために、ループカウンタは異なる値、例えば“2”や“5”をロードしてある。

【００８５】シングルループの複雑でない実行と対照すると、ネスティングはより多くのレジスタ、ネスティングスタック、付加的制御論理を必要とする。

【００８６】ネスティングされた複数のループ処理の各開始アドレスはレジスタＲＳ１（Ｓｔａｒｔ＿1）～ＲＳｎ（Ｓｔａｒｔ＿n）（n≧2）に記憶され、これらの出力はセレクタ（ＳＥＬ＿ＳＴＡＲＴ）４にそれぞれ付与される。また、各終了アドレスはレジスタＲＥ１（Ｅｎｄ＿1）～ＲＥｎ（Ｅｎｄ＿n）に記憶され、これらの出力はセレクタ（ＳＥＬ＿ＥＮＤ）５にそれぞれ付与される。

【００８７】セレクタ４は、選択制御信号ｔ１＿ｃｏｎｔに基づき、レジスタＲＳ１～ＲＳｎの格納データのうち一のデータを選択開始アドレスとしてレジスタ７に出力する。セレクタ５は、選択制御信号ｔ１＿ｃｏｎｔに基づきレジスタＲＥ１～ＲＥｎの格納データのうち一のデータを選択終了アドレスとしてレジスタ８に出力する。

【００８８】各ループ処理の反復数は、各レジスタＲＬ１（Ｌｏｏｐ＿1）～レジスタＲＬｎ（Ｌｏｏｐ＿n）にそれぞれ記憶され、これらのレジスタＲＬ１～ＲＬｎの出力はセレクタ６（ＳＥＬ＿ＬＯＯＰ）に付与される。セレクタ６は、選択制御信号ｔ１＿ｃｏｎｔに基づき、レジスタＲＬ１～ＲＬｎの格納データのうち一のデータを選択反復数としてセレクタ２４に出力する。

【００８９】タイミングに関しては、レジスタＲＬ１～ＲＬｎ、ＲＳ１～ＲＳｎ及びＲＥ１～ＲＥｎはクロックｔ１に同期してラッチする。レジスタ７及び８はクロックｔ１に同期してラッチする。

【００９０】レジスタ７、レジスタ８の出力からは、図２及び図３で示した第１の実施例のプログラム制御装置と同様な回路構成となる。

【００９１】可変ループカウンタ２６Ｌ１～２６Ｌｎもまた、マルチレベル・ネスティングに合わせてｎ個設けられる。

【００９２】ループの階層構造は、可変ループカウンタ２６Ｌ１～２６Ｌｎをそれぞれマルチプレクシングしたりデマルチプレクシングすることによって、単純化を維持される。従って、セレクタ２５（ＬＣ＿ＩＮ）は、セレクタ２４（ＳＥＬ＿ＬＣ）の出力をデマルチプレクシングし、その出力データをクロックｔ２に同期してループカウンタ２６Ｌ１（ＬＣ＿1）～２６Ｌｎ（ＬＣ＿n）にラッチさせる。セレクタ２７（ＬＣ＿ＯＵＴ）は各可変ループカウンタ２６Ｌ１～２６Ｌｎの出力をマルチプレクシングし、その出力データをクロックｔ１に同期してレジスタ２８（ｔ１＿１ｃ）にラッチさせる。

【００９３】レベルインジケータ３３は、クロックｔ１及びｔ２並びにコンパレータ１２及び１３からそれぞれ出力されるcmp_start及びcmp_endを受け、選択制御信号ｔ１＿ｃｏｎｔ及び選択制御信号ｔ２＿ｃｏｎｔを出力する。表３にレベルインジケータ３３の制御体系を現す真理値表を示す。なお、選択制御信号ｔ１＿ｃｏｎｔ及び選択制御信号ｔ２＿ｃｏｎｔのデフォルト値は“0”である。ただし、cmp_endはcmp_startよりクロックｔ２（ｔ１）の１サイクル前の値を意味する。

【００９４】

【表３】

15　　16

cmp_start	cmp_end {delayed for 1 cycle!!!}	Function
0	0	Hold
1	0	Increment
0	1	Decrement
1	1	Hold

【００９５】表３に示すように、選択制御信号ｔ１＿ｃｏｎｔ及びｔ２＿ｃｏｎｔは、cmp_start及びcmp_endが共に“０”、共に“１”である場合は現在の状態が維持され、cmp_startが“１”でcmp_endが“０”である場合はインクリメントされ、cmp_startendが“０”でcmp_endが“１”である場合はデクリメントされる。

【００９６】セレクタ４～６及び２７は、レベルインジケータ３３からの選択制御信号ｔ１＿ｃｏｎｔに基づき選択動作を行う。セレクタ５、６及び２７の選択体系を表４に示し、セレクタ４の選択体系を表５に示す。

【００９７】

【表４】

t1_cont	Action
0	？？？＿１ --> Output
1	？？？＿１ --> Output
2	？？？＿２ --> Output
・・・	・・・
n	？？？＿ｎ --> Output

？？？＝ＥＮＤ、ＬＯＯＰｏｒＬＣ

【００９８】

※　※【表５】

t1_cont	Action
0	Start_1 --> Output
1	Start_2 --> Output
2	Start_3 --> Output
・・・	・・・
n－１	Stsrt_n --> Output
n	' 0' --> Output

【００９９】表４に示すように、セレクタ５、６及びセレクタ２７は、それぞれ選択制御信号ｔ１＿ｃｏｎｔが“０”のときレジスタＲＥ１，レジスタＲＬ１及び可変ループカウンタ２６Ｌ１の格納データを出力し、選択制御信号ｔ１＿ｃｏｎｔがｉ（ｉ＝１～ｎ）のときレジスタＲＥｉ，レジスタＲＬｉ及び可変ループカウンタ２６Ｌｉの格納データを出力する。

【０１００】表５に示すように、セレクタ４は、選択制御信号ｔ１＿ｃｏｎｔがｊ（ｊ＝０～（ｎ－１））のときレジスタＲＳ（ｊ＋１）の格納データを出力し、選択制御信号ｔ１＿ｃｏｎｔがｎのとき“０”を出力する。

【０１０１】セレクタ２５は、レベルインジケータ３３の選択制御信号ｔ２＿ｃｏｎｔに基づき選択動作を行う。セレクタ２５の選択体系を表６に示す。

【０１０２】

【表６】

17　　18

t2_cont	Action
0	LC_1 --> Output
1	LC_1 --> Output
2	LC_2 --> Output
...	...
n	LC_n --> Output

【0103】表6に示すように、セレクタ25は、選択制御信号t2_contが“0”のとき可変ループカウンタ26L1にデータを出力し、選択制御信号t2_contがi（i＝1～n）のとき可変ループカウンタ26Liにデータを出力する。

【0104】デフォルトにより、選択制御信号t1_cont及び選択制御信号t2_contが“0”に設定されているため、シングルループをエミュレートする。

【0105】図11～図13の波形図に示すように、選択制御信号t1_cont及び選択制御信号t2_contにより、ループのネスティングの現在のレベルが現され、選択制御信号t1_contに基づき、レジスタRL1～RLn、レジスタRE1～REn及びレジスタRS1～RSnの格納データがセレクタ4、5及び6によりそれぞれ選択される。

【0106】同様に、可変ループカウンタ26L1～26Lnの格納データについては、慎重な対処を必要とするため、セレクタ25は選択制御信号t2_contに基づき、セレクタ24の出力を可変ループカウンタ26L1～26Lnの一に格納し、セレクタ27は選択制御信号t1_contに基づき可変ループカウンタ26L1～26Lnの格納データのうち一のデータをレジスタ28に出力する。

【0107】このように、選択制御信号t1_cont及び選択制御信号t2_contに基づき、セレクタ4～6、25及び27により、レジスタRS1～RSn、レジスタRE1～REn、レジスタRL1～RLn及び可変ループカウンタ26L1～26Lnの格納データのマルチプレクシングおよびデマルチプレクシングすることにより、ネスティング構造のループ命令にも対応することができる。

【0108】なお、それ以外の構成及び動作は、第1の実施例のプログラム制御装置と同様であるため、説明を省略する。

【0109】このように、第2の実施例のプログラム制御装置は、レベルインジケータ33からの選択制御信号t1_cont及び選択制御信号t2_contにより、現在制御対象となるループ処理が指示され、プログラムカウンタと選択開始アドレスとの比較結果に基づき、制御対象のループ処理の開始を認識することができるため、プログラム中において第1～第nのループ命令は第1～第nのループ実行命令群の前であれば任意の位置に記述することができる。

【0110】例えば、図14で示したように、第1のループ実行命令群であるブロックBlock1～3及び第2のループ実行命令群であるブロックBlock2の前の行番号21に第1のループ命令「loop lc1, Start_1, End_1」を記述し、行番号23に第2のループ命令「loop lc2, Start_2, End_2」を記述することができる。

【0111】したがって、図18で示したプログラムのように、第1及び第2のループ実行命令群からなるネスティング構造中に前記第1及び第2のループ命令を記述することによる不具合を解消することができ、第1のループ命令の反復回数に関係なく第2のループ命令は1回のみ実行される。

【0112】その結果、第1～第nのループ実行命令群によるネスティング構造の外に第1～第nのループ命令を記述することにより、ネスティング構造の第1～第nのループ処理を効率的に行うことができる。

【0113】

【発明の効果】以上説明したように、この発明における請求項1記載のプログラム制御装置のループ制御手段は、プログラムカウンタと開始アドレスとの比較結果に基づきループ処理の開始を認識することができるため、プログラム中においてループ命令はループ実行命令群の前であれば任意の位置に記述することができる。

【0114】その結果、ループ命令をループ処理の実行の妨げにならない箇所に記述することにより、効率的なループ処理を行うことができる。

【0115】この発明における請求項2記載のプログラム制御装置は、プログラムカウンタと選択開始アドレスとの比較結果及びプログラムカウンタと選択終了アドレスとの比較結果に基づき、現在制御対象となるループ処理を指示する選択制御信号を出力する選択制御信号出力手段と、プログラムカウンタと選択開始アドレスとの比較結果に基づき、選択開始アドレスに対応するループ処理の開始を認識し、プログラムカウンタと選択終了アドレスとの比較結果に基づき選択終了アドレスに対応するループ処理の終了を認識して、制御対象のループ処理が前記選択反復数回、繰り返して実行されるようにプログラムカウンタを設定するループ制御手段とを備えてい

19

る。

【０１１６】したがって、プログラム中において、第１～第ｎのループ命令はそれぞれ第１～第ｎのループ実行命令群の前であれば任意の位置に記述することができる。

【０１１７】その結果、第１～第ｎのループ実行命令群によるネスティング構造の外に第１～第ｎのループ命令を記述することにより、ネスティング構造の第１～第ｎのループ処理を効率的に行うことができる。

【０１１８】この発明における請求項３記載のプログラム制御方法のステップ(c)は、プログラムカウンタと開始アドレスとの比較結果に基づきループ処理の開始を認識し、プログラムカウンタと終了アドレスとの比較結果に基づきループ処理の終了を認識することにより、ループ実行命令群の命令が反復数回、繰り返して実行されるようにプログラムカウンタを設定している。

【０１１９】したがって、プログラム中においてループ命令はループ実行命令群の前であれば任意の位置に記述することができる。

【０１２０】その結果、ループ命令をループ処理の実行の妨げにならない箇所に記述することにより、効率的なループ処理を行うことができる。

【図面の簡単な説明】

【図１】　この発明の第１の実施例のプログラム制御装置で実行制御されるプログラム例を示す説明図である。

【図２】　この発明の第１の実施例であるプログラム制御装置の構成を示す回路図である。

【図３】　この発明の第１の実施例であるプログラム制御装置の構成を示す回路図である。

【図４】　第１の実施例のプログラム制御装置の動作を示す波形図である。

【図５】　第１の実施例のプログラム制御装置の動作を示す波形図である。

【図６】　第１の実施例のプログラム制御装置の動作を示す波形図である。

20

【図７】　この発明の第１の実施例のプログラム制御装置で実行制御されるプログラム例を示す説明図である。

【図８】　この発明の第２の実施例のプログラム制御装置で実行制御されるプログラム例を示す説明図である。

【図９】　この発明の第２の実施例であるプログラム制御装置の構成を示す回路図である。

【図１０】　この発明の第２の実施例であるプログラム制御装置の構成を示す回路図である。

【図１１】　第２の実施例のプログラム制御装置の動作を示す波形図である。

【図１２】　第２の実施例のプログラム制御装置の動作を示す波形図である。

【図１３】　第２の実施例のプログラム制御装置の動作を示す波形図である。

【図１４】　この発明の第２の実施例のプログラム制御装置で実行制御されるプログラム例を示す説明図である。

【図１５】　プログラム実行システムの全体構成を示す説明図である。

【図１６】　第１の実施例のプログラム制御装置と等価なプログラム制御方法を示すフローチャートである。

【図１７】　従来のプログラム制御装置で実行制御されるプログラム例を示す説明図である。

【図１８】　従来のプログラム制御装置で実行制御されるプログラム例を示す説明図である。

【図１９】　画素のマスク処理例を示す説明図である。

【図２０】　従来のプログラム制御装置で実行制御されるプログラム例を示す説明図である。

【符号の説明】

７～９　レジスタ、１１，１２　コンパレータ、１９，２０　論理回路、ＲＳ１～ＲＳｎ　レジスタ、ＲＥ１～ＲＥｎ　レジスタ、ＲＬ１～ＲＬｎ　レジスタ、２６Ｌ１～２６Ｌｎ　可変ループカウンタ、３３　レベルインジケータ。

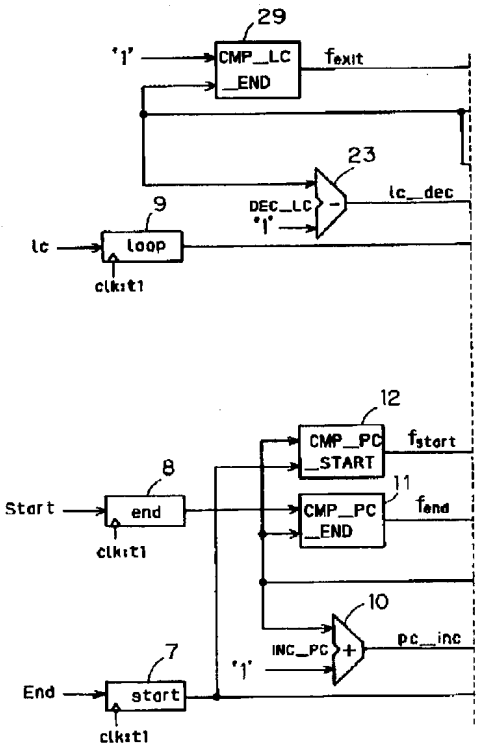
【図１】

```
...:>          ...
 20:>      load 3,ic;
 21:>      loop ic,Start,End;
...:>          ...
 40:> Start:  {First Instruction}   ┐
 41:>          ...                  ├ B1
 42:> End:    {Last Instruction}    ┘
...:>          ...
```

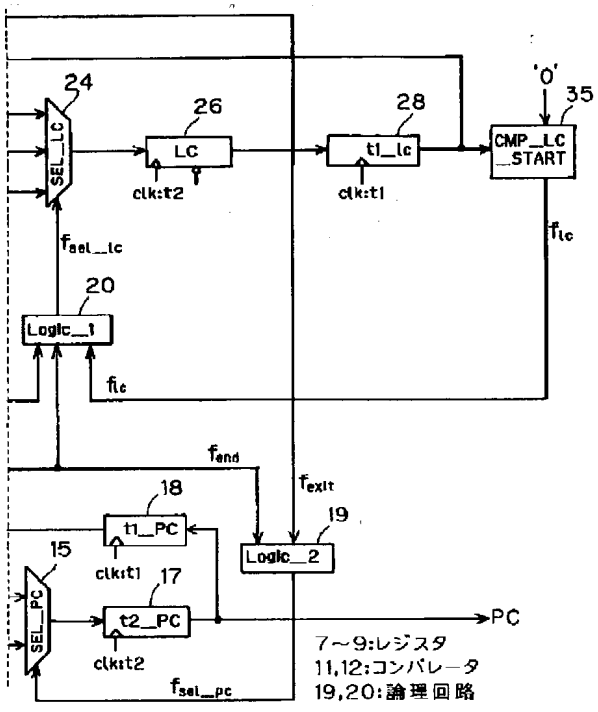
【図７】

```
20:>      load 2,lc;
21:>      loop lc,Start,End;
22:>          ...
37:>      mulacc a[0],0;              ┐
38:>      mulacc a[0],wr[0];          ├ BlockA
39:>      mulacc a[1],wr[1];          ┘
40:> Start: mulacc a[i,1],wr[0];      ┐
41:>      mulacc a[i,1],wr[1];        ├ BlockB[i]
42:> End:   mulacc a[i,-2],wr[2];     ┘
43:>      mulacc a[i,1],wr[0];        ┐
44:>      mulacc a[i,1],wr[1];        ├ BlockC
45:>      mulacc a[i],0;              ┘
```

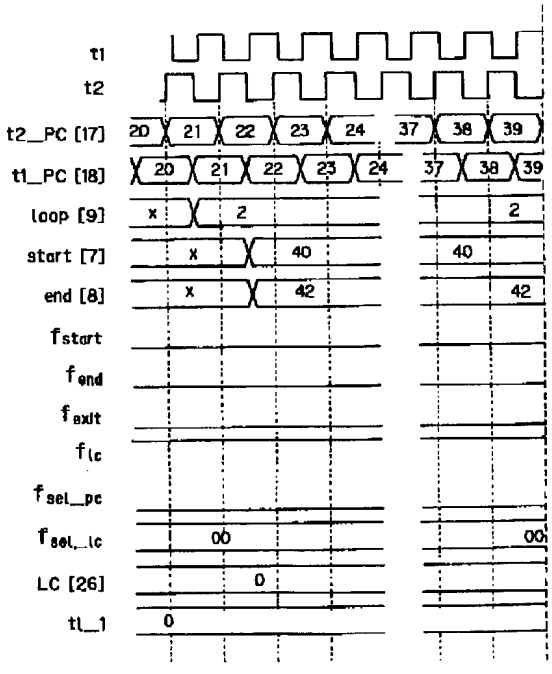
【図2】


29
'1'
CMP_LC _END
fexit
23
DEC_LC
'1'
lc_dec
9
lc
loop
clk:t1
12
CMP_PC _START
fstart
8
Start
end
clk:t1
11
CMP_PC _END
fend
10
INC_PC
'1'
pc_inc
7
End
start
clk:t1

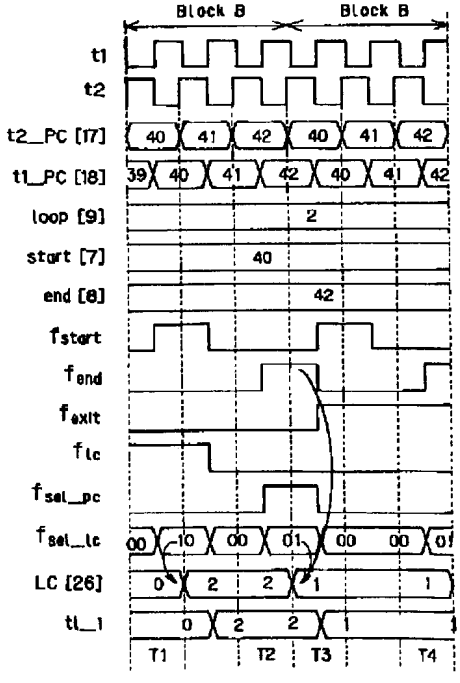
【図3】


24
SEL_LC
26
LC
clk:t2
28
t1_lc
clk:t1
'0'
35
CMP_LC _START
flc
fsel_lc
20
Logic_1
flc
fend
fexit
18
t1_PC
clk:t1
19
Logic_2
15
SEL_PC
17
t2_PC
clk:t2
PC
fsel_pc
7～9:レジスタ
11,12:コンパレータ
19,20:論理回路

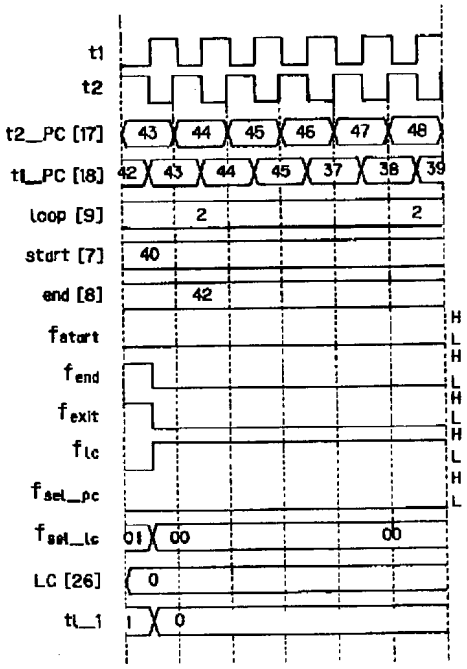
【図4】


t1
t2
t2_PC [17]
t1_PC [18]
loop [9]
start [7]
end [8]
fstart
fend
fexit
flc
fsel_pc
fsel_lc
LC [26]
t1_1

【図5】


Block B
Block B
t1
t2
t2_PC [17]
t1_PC [18]
loop [9]
start [7]
end [8]
fstart
fend
fexit
flc
fsel_pc
fsel_lc
LC [26]
t1_1
T1
T2
T3
T4

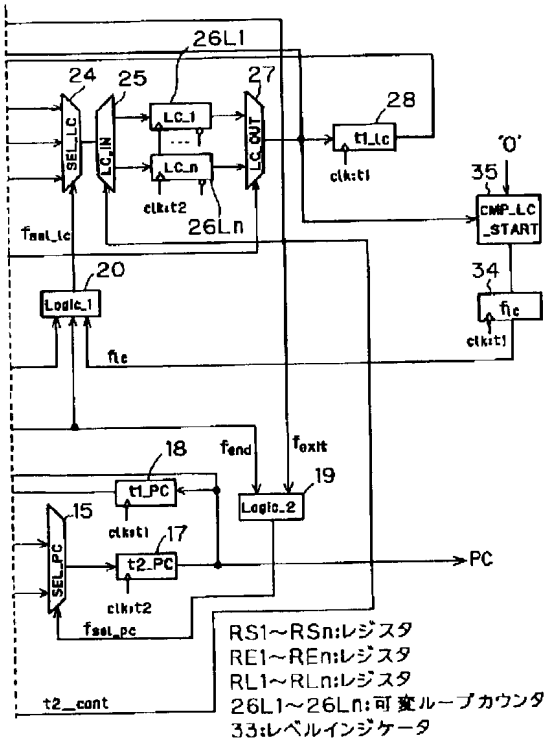
【図6】



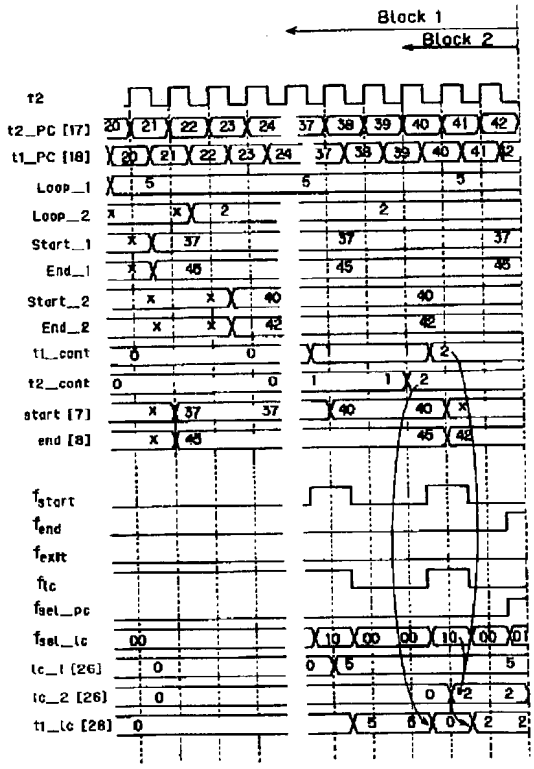
【図8】

```
...:>              ...
20:>          load3,lc1;
23:>          loop lc1,Start_1,End_1
22:>          load 3,lc2;
23:>          loop lc2,Start_2,End_2
...:>
37:> Start_1: {Firast Instruction(Block1)}
...:>
40:> Start_2: {Firast Instruction(Block2)}
...:>              ...                          B12  B11
42:> End_2:       {Last Instruction(Block2)}
...:>              ...
45:> End_1:       {Last Instruction(Block1)}
...:>              ...
```

【図10】



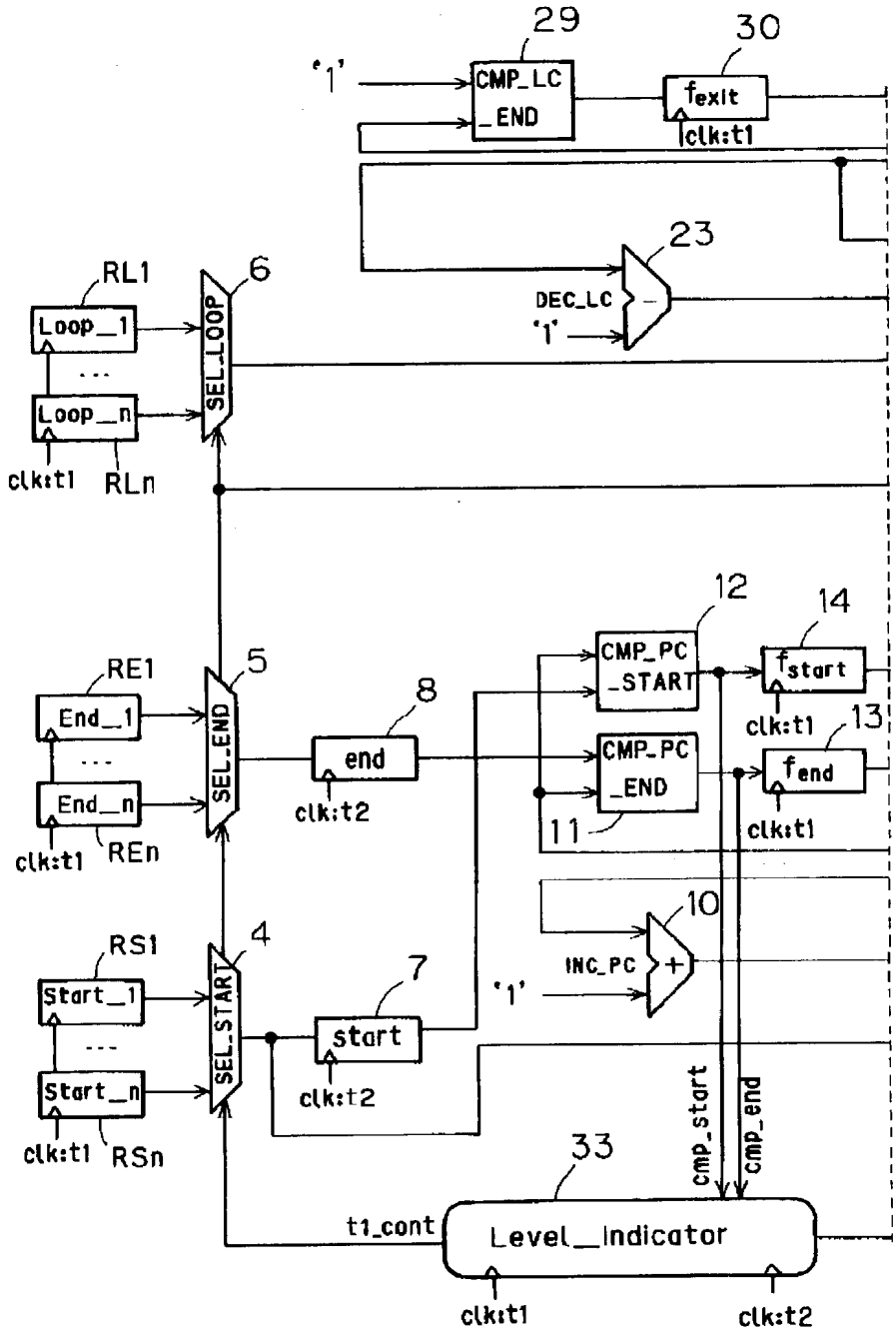
【図11】



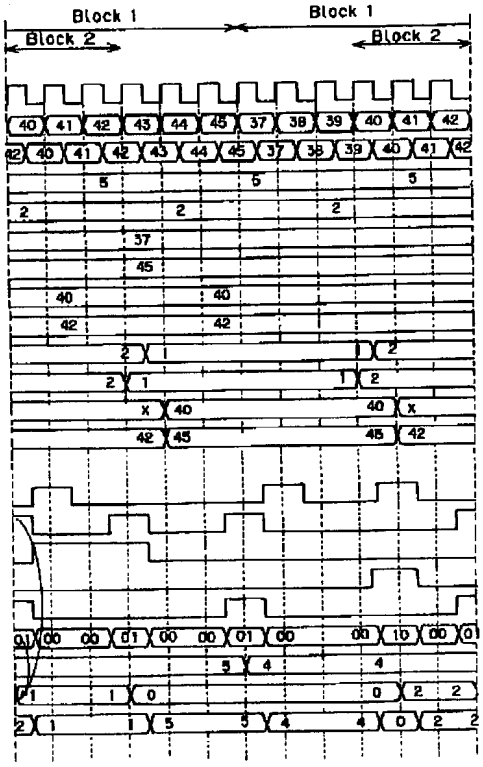
【図17】

```
...:>              ...
38:>          load3,lc;
39:>          loop lc,End
40:>          {First Indstruction}
...:>
42:> End:     {Last Instruction}
43:>               ...
```

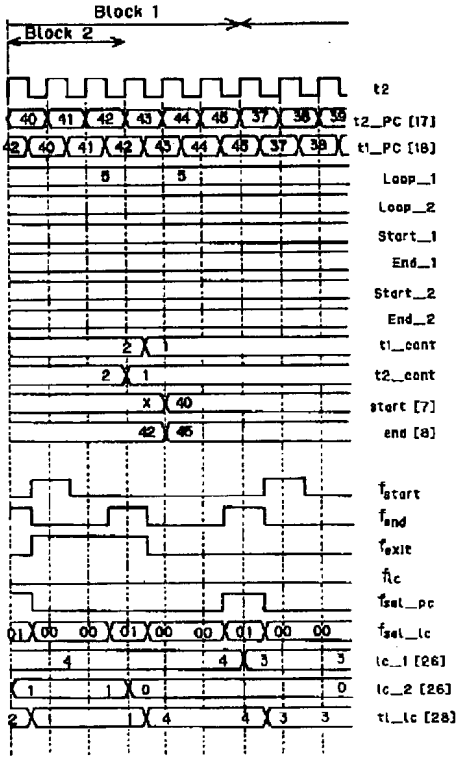
【図9】


29
30
'1'
CMP_LC
_END
fexit
clk:t1
23
DEC_LC
'1'
RL1
6
Loop_1
Loop_n
SEL_LOOP
clk:t1
RLn
12
14
CMP_PC
_START
fstart
clk:t1
RE1
5
8
13
End_1
End_n
SEL_END
end
clk:t2
CMP_PC
_END
fend
clk:t1
clk:t1
REn
11
10
INC_PC
'1'
RS1
4
7
Start_1
Start_n
SEL_START
start
clk:t2
clk:t1
RSn
cmp_start
cmp_end
33
t1_cont
Level_Indicator
clk:t1
clk:t2

【図12】



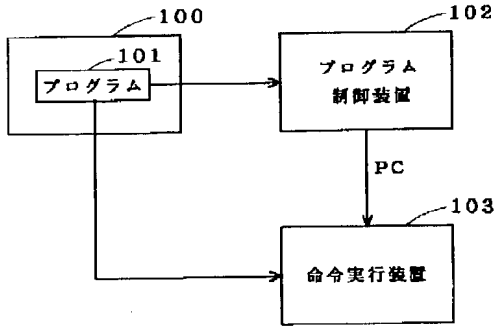
【図13】



【図14】

```
...:>              ...
20:>           load 5,lc1;
21:>           loop lc1,Start_1,End_1;
22:>           load 2,lc2;
23:>           loop lc2,Start_2,End_2;
...:>              ...
37:> Start_1:  mulacc a[0],0;          ┐
38:>           mulacc a[0],wr[0];      │ Block1
39:>           mulacc a[1],wr[1];      ┘
40:> Start_2:  mulacc a[i,1],wr[0];    ┐
41:>           mulacc a[i,1],wr[1];    │ Block2
42:> End_2:    mulacc a[i,-1],wr[2];   ┘
43:>           mulacc a[i,1],wr[0];    ┐
44:>           mulacc a[i,1],wr[1];    │ Block3
45:> End_1:    mulacc a[i],0;          ┘
...:>              ...
```

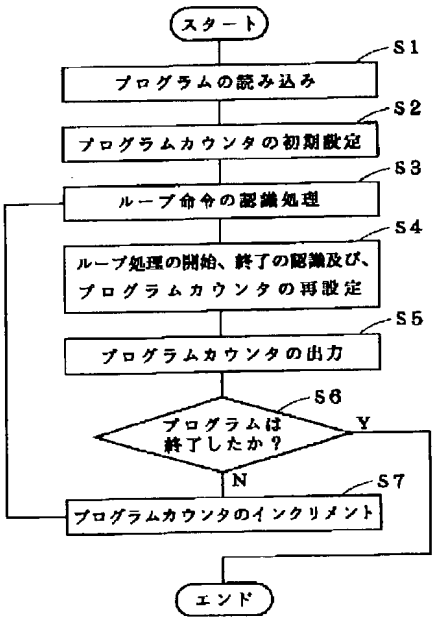
【図15】



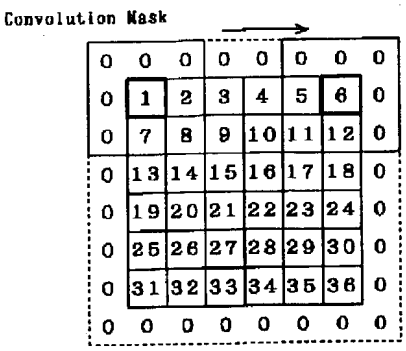
【図18】

```
...:>              ...
33:>           load 3,lc1;
34:>           load 3,lc2;
35:>          loop lc1,End_1
36:>          {First Instruction(Block2)}   ┐
...:>              ...                       │
39:>          loop lc2,End_2                 │
40:>          {First Instruction(Block2)}    │
41:>              ...                        │
42:> End_2:   {Last Instruction(Block2)}     ┘
...:>              ...
45:> End_1:   {Last Instruction(Block1)}     ┘
46:>              ...
```

【図１６】



【図１９】



【図２０】

```
Line Label Instruction

35:>          load 4,lc1;
36:>          mulacc a[0],0;          ┐
37:>          mulacc a[0],wr[0];      ┘ BlockA
38:>          mulacc a[1],wr[1];
39:>          loop  lc1,End;          ← Loop Instruction
40:>          mulacc a[i,1],wr[0];    ┐
41:>          mulacc a[i,1],wr[1];    ┘ BlockB[i]
42:> End:     mulacc a[i,1],wr[2];
43:>          mulacc a[i,1],wr[0];    ┐
44:>          mulacc a[i,1],wr[1];    ┘ BlockC
45:>          mulacc a[i],0;
```

フロントページの続き

(72)発明者　井上　喜嗣
兵庫県伊丹市瑞原４丁目１番地　三菱電機
株式会社システムエル・エス・アイ開発研
究所内